Some practical aspects of model predictive control

Ruth Bars[a], Robert Haber[b], Ulrich Schmitz[b], Zsuzsa Preitl[a]

[a]Dept. of Automation and Applied Informatics, Budapest Univ. of Technology & Economics, HAS-BME, Control Research Group, Hungary;

[b]Dept. of Plant and Process Engineering, Univ. of Applied Science, Cologne, Germany

ABSTRACT

Predictive control algorithms calculate a series of the control signal values minimizing a quadratic cost function evaluating the difference between the reference signal and the predicted process output in a given future horizon. Minimizing the cost function constraints can be considered. With predictive control strategy better control performance can be achieved than with the usual PID control algorithms especially in case of known reference values and significant dead times. Optimization considering constraints can be a time consuming operation especially in case of long prediction horizon. An important practical problem is decreasing the calculations executed between the sampling points in real time applications. The paper presents some possibilities to reduce the computation time. It is shown that exponential allocation of the considered points in the prediction horizon instead of equidistant allocation and the appropriate choice of the allowed change points in the control signal may provide a drastic decrease in computation time. Another approach to decrease the calculations is using the Generalized Predictive Control (GPC) algorithm for unconstrained case and to handle the limitations set for the control signal afterwards. For the case of stable plants GPC is transformed into internal model control (IMC) structure. An effective aposteriori constraint handling is shown applying the controller dynamics in the feedback of the static saturation.

Keywords: predictive control algorithms, coincidence points, blocking technique, IMC form of GPC, constraint handling

1. INTRODUCTION

Predictive control algorithms calculate the actual and forthcoming control signal values in a given horizon minimizing a quadratic cost function which evaluates the difference between the reference signal and the predicted process output in a given future horizon. The future values of the reference signal are given in a lot of applications (e.g. in robot control). The future values of the plant can be predicted from the model of the plant. The future output is affected partly by the past inputs, and partly by the actual and future inputs which are optimized. From the calculated control signal series only the first value is applied at the input of the process and in the next sampling point the calculation is repeated (receding horizon strategy). Minimizing the cost function constraints can be considered. The tuning parameters of the algorithm are the starting point and the length of the prediction horizon, the control horizon (the supposed executive changes in the control signal) and the weighting factors in the cost function.

The different predictive control algorithms differ mainly in the process model (linear or nonlinear, parametric or nonparametric).

With predictive control strategy better control performance can be achieved than with usual control algorithms especially in case of known reference values and significant dead times. Receding horizon strategy will ensure appropriate disturbance rejection and will decrease the sensitivity to plant-model mismatch.

Besides the PID control algorithms predictive control algorithms have reached a wide industrial acceptance (Qin, 2003). This trend is increasing as effective industrial program packages are available for predictive control.

The control algorithm minimizes the cost function

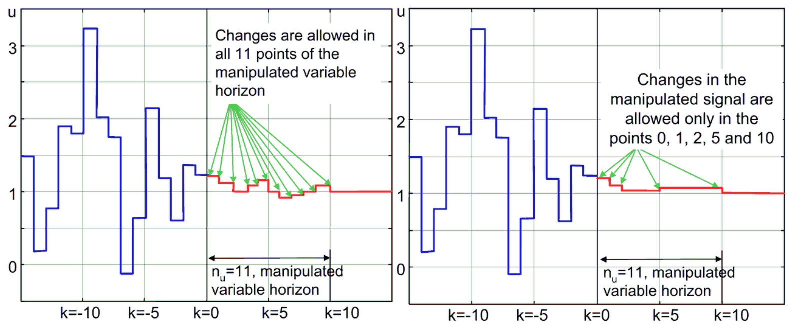
$$J = \sum_{n_e=n_{e1}}^{n_{e2}} \left[y_{ref}(k+n_e+d) - \hat{y}(k+n_e+d)\right]^2 + \lambda_u \sum_{j=1}^{n_u} \Delta u(k+j-1)^2 \Rightarrow \min_{\Delta u} \tag{1}$$

u is the input, y is the output signal of the plant, y_{ref} denotes the future reference signal. n_{e1} is the starting point, and n_{e2} is the end point of the control error horizon. n_u denotes the length of the manipulated variable horizon, which is also equal to the number of maximal allowed changes of the manipulated variable. d is the mathematical dead time and λ_u indicates the weighting factor for the manipulated signal increments,

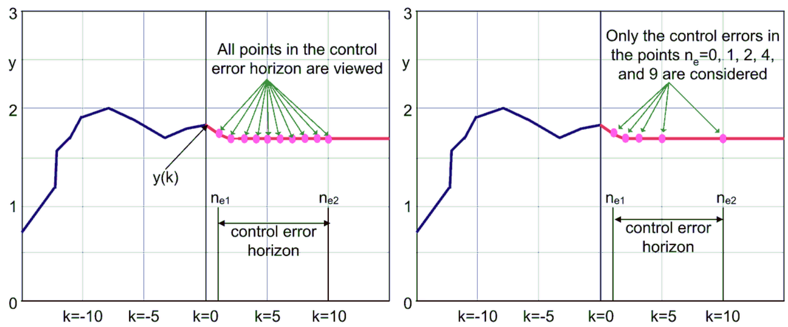
An important practical problem is reducing the calculations which have to be executed between the sampling points. Optimization considering constraints can be a time consuming operation especially in case of long prediction horizon. This problem becomes critical with small sampling periods. Real time applications would require techniques reducing the computation time significantly. The paper considers some possibilities to reduce the computation time of predictive control algorithms.

2. SAVING COMPUTATIONAL EFFORT USING COINCIDENCE POINTS AND BLOCKING TECHNIQUE

Optimization considering constraints requires significant time if it has to be executed in a big number of points in the prediction horizon. Using so called blocking and coincidence points techniques can reduce the computation time while keeping the length of the horizon. With blocking technique, changes in the control (manipulated) variable are supposed only in given predefined points during the calculation. Coincidence points mean considering a number of appropriately located points in the prediction (control error) horizon. This way the length of the prediction horizon is kept while the number of points where the calculation is executed can be reduced significantly. Fig. 1. demonstrates the conception of blocking technique and coincidence points. Using both techniques significant reduction of the number of the used points is possible; a reduction to only 10% of the original number of points is not unusual.



(a) blocking technique



(b) coincidence points technique

Figure 1. Coincidence points and blocking technique

The reduction of the computational effort is only nearly linear with the number of the reduction of the points, as the computational effort of prediction of points settled far in the future is more extensive than the computation of the points near to the actual point. This behavior can be demonstrated by the example of polynomial process models in their predictive form.

Applied to a very simple example of a plant given by its pulse transfer function $P(q^{-1}) = \frac{0.5q^{-1}}{1-0.5q^{-1}}$ and horizon length of 4 points with $n_{e1} = 0$ and ne2=3 the prediction equations are given by:

$$\begin{array}{rcllllll}
\hat{y}(k+1|k) & = & 0.5y(k) & & & & +0.5u(k) \\
\hat{y}(k+2|k) & = & 0.25y(k) & & & +0.5u(k+1) & +0.25u(k) \\
\hat{y}(k+3|k) & = & 0.125y(k) & & +0.5u(k+2) & +0.25u(k+1) & +0.125u(k) \\
\hat{y}(k+4|k) & = & 0.0625y(k) & +0.5u(k+3) & +0.25u(k+2) & +0.125u(k+1) & +0.0625u(k)
\end{array} \quad (2)$$

It is clear that the computational effort for the calculation of the points which are far from the actual point in the future (here at $k + 4$) is higher than the computational effort necessary for the computation of the points close to the actual time point (here at $k + 1$).

Nevertheless it is still an unanswered question how to allocate these points in a given horizon. Many users of industrial predictive control have some heuristic experience on how to allocate the points in the horizons but beyond some simple rules of thumb no instructions on these techniques are known. Here a possible solution is presented applying a two-stage optimization process, using a genetic optimizer to find the parameters for the best possible control for a SISO (Single-Input, Single-Output) predictive controller.

The most important parameters of predictive control are the horizons (n_{e1}, n_{e2}, and n_u) and the weighting factor λ_u. The horizons are characterized by integer values. A genetic optimizing algorithm has been used to minimize a specified cost function, because it is able to handle integer variables as the optimization variable.

The optimization is done in a two-stage process:

1. The length of the control error and manipulated variable horizon were optimized without reducing the computation points in the horizons,
2. The allocation of the computation points in both horizons were optimized so that the controlled signal should approximate the controlled signal without reducing the computation points in the horizons.

Choice of the horizons

As most processes in industry can be approximated by an aperiodic second order process, such processes were used as basis for the optimization approach. A number of processes have been investigated, all containing a bigger and a smaller time constant between 1 and $30sec.$ (T_{small} and T_{big}). In all cases the sampling time was $\Delta T = 0.2sec.$

The length of the horizons was optimized according to three criteria in case of a reference step input:

- the settling time,
- the maximum overshoot,
- the length of the horizons.

The third criterion was taken into consideration to prevent the optimization algorithm to make the horizons longer than necessary in order to spare computation time. Therefore, the lengths of the horizons were considered also, but only if they are longer than a useful value:

- control error horizon length: four times the sum of the time constants ($n_{e2,max} > 4T_\Sigma$)
- manipulated signal horizon length: the sum of the time constants ($n_{u,max} > T_\Sigma$)

The optimization itself was performed by means of the genetic optimization algorithm written by (Sekaj, 2002). The cost function to be optimized was in all cases

$$\begin{aligned} J &= \overbrace{\lambda_{set}\cdot\ t_{98\%}}^{J_{set}} + \overbrace{\lambda_{of}\cdot|y_{max}-y_{ref}|}^{J_{ov}} + \overbrace{\lambda_{e,hor}\cdot V_{ne}+\lambda_{u,hor}\cdot V_{nu}}^{J_h} \\ &\Rightarrow \min_{n_{e1},n_{e2},n_u,\lambda_u} \end{aligned} \quad (3)$$

$$\text{with}\quad V_{ne}=\sum_{i=n_{e1}}^{n_{e2}}\begin{cases}0 \text{ for } d+1+i\le 4\cdot\frac{T_\Sigma}{\Delta T}\\ i \text{ for } d+1+i> 4\cdot\frac{T_\Sigma}{\Delta T}\end{cases}$$

$$\text{and}\quad V_{nu}=\sum_{i=1}^{n_u}\begin{cases}0 \text{ for } i-1\le \frac{T_\Sigma}{\Delta T}\\ i \text{ for } i-1> \frac{T_\Sigma}{\Delta T}\end{cases}$$

In the simulation all weighting factors were set to $\lambda_{98\%}=\lambda_{ov}=\lambda_{e,hor}=\lambda_{u,hor}=1$.

On the basis of the optimal results obtained for a number of processes the following rules of thumb can be given for the optimal choice of the horizons:

- The beginning of the control error horizon should be 4% of the big time constant, thus: $n_{e1}=0.04\cdot(\frac{T_{big}-T_{small}}{\Delta T})$
- The end of the control error horizon should be nearly: $n_{e2}=\begin{cases}-4\cdot(\frac{T_{big}}{T_{small}}+19) & \text{for } \frac{T_{big}}{T_{small}}\ge 10\\ (\frac{1}{3})\cdot(\frac{T_{big}+T_{small}}{\Delta T}+150) & \text{for } \frac{T_{small}}{T_big}<10\end{cases}$
- The length of the manipulated variable horizon should be $n_u=\begin{cases}0.045\cdot\frac{T_{big}+T_{small}}{\Delta T} & \text{for } \frac{T_{big}}{T_{small}}\ge 10\\ 0.055\cdot\frac{T_{big}+T_{small}}{\Delta T} & \text{for } \frac{T_{big}}{T_{small}}<10\end{cases}$

Reducing the number of points within the horizons

The next task was to reduce the large amount of point to $P_e=20$ for the control error horizon and $P_u=10$ for the manipulated variable horizon.

The aim of the optimization process was to achieve a control with the reduced number of points that is nearest to the control with the optimized size of the horizon but without blocking or using coincidence points. The cost function to be optimized is described by

$$\begin{aligned} J &= J_e+J_u+J_h \\ &= +\lambda_{e,dev}\cdot\sum_{k=0}^{200}(y_{full}(k)-y_{red}(k))^2 \\ &\quad +\lambda_{u,dev}\cdot\sum_{k=0}^{200}(u_{full}(k)-u_{red}(k))^2 \\ &\quad +\lambda_{Pe}\cdot V_{Pe}+\lambda_{Pu}\cdot V_{Pu} \\ &\Rightarrow \min_{P_e,P_u} \end{aligned} \quad (4)$$

$$\text{with}\quad V_{Pe}=\sum_{i=1}^{N_{B,n_e}}\begin{cases}0 \text{ for } P_i\le 4\cdot T_\Sigma\\ i \text{ for } P_i> 4\cdot T_\Sigma\end{cases}$$

$$\text{and}\quad V_{Pu}=\sum_{i=1}^{N_{B,n_u}}\begin{cases}0 \text{ for } P_i\le T_\Sigma\\ i \text{ for } P_i> T_\Sigma\end{cases}$$

with

y_{full} controlled variable, controlled with all points in the horizons

y_{red} controlled variable, controlled with a reduced number of points in the horizons

u_{full} manipulated variable with all points in the horizons

u_{red} manipulated variable with a reduced number of points in the horizons

P_e allocation of the points in the control error horizon

λ_{P_e} weighting factor for the allocation of the points in the control error horizon

P_u allocation of the points in the manipulated variable horizon

λ_{P_u} weighting factor for the allocation of the points in the manipulated variable horizon

$\frac{T_\Sigma}{\Delta T}$ sum of all time constants of the process relative to the sampling time

$\lambda_{y,dev}$ weighting factor for the error between the controlled variables with and without reduced horizons, here chosen to be $\lambda y, dev = 1$

$\lambda_{u,dev}$ weighting factor for the error between the manipulated variable with and without reduced horizons, here chosen to be $\lambda u, dev = 0.1$

N_{B,n_e} is the (reduced) number of points in the control error horizon

N_{B,n_u} is the (reduced) number of points in the manipulated variable increment horizon

Fig. 2. demonstrates the control result of the genetic optimization process by solid lines. The step and impulse responses of a considered sample second order system are given. The dashed lines show the performance with the reduced number of points, the full lines give the original outputs. The reference signal is given by dash-dotted line. Fig. 3. shows the location of the optimal coincidence points in the step response and in the impulse response of the corresponding process.

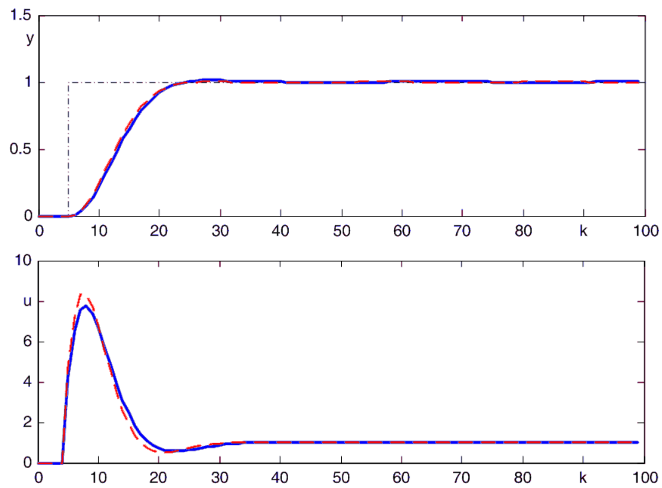


Figure 2. Control with the optimal allocation of the reduced number of points in the horizons for a second order sample process

The optimal blocking points can be also determined. From the investigations it can be concluded that the best result can be achieved if the coincidence points are settled at the rising part of the impulse response of the process. Also it can be seen that the coincidence points should be more on the rising part of the impulse

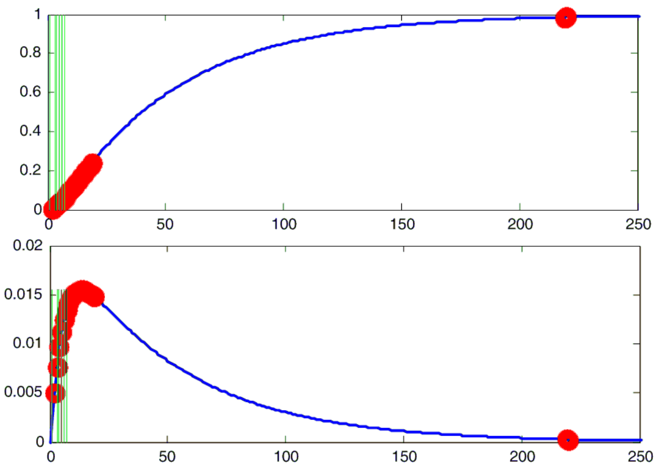


Figure 3. Optimal allocation of the coincidence points for the sample process

response if the ratio between the big and the small time constant becomes bigger. On the other hand, if the ratio between the time constants is lower, the points are located both on the rising part of the impulse response and also at the maximum and on the early part of the falling part of the impulse response. This behavior may be founded in the fact that the bigger the ratio between the time constants of a second-order process is, the more the process can be approximated by a first-order process. In addition to this some few points are settled at the very end of the step response, in a part of the response, where the process acts mostly static to the change in the manipulated signal.

3. IMC REPRESENTATION OF UNCONSTRAINED GPC CONTROL WITH APOSTERIORI CONSTRAINT HANDLING

Another approach to decrease the calculations is using the Generalized Predictive Control (GPC) algorithm for unconstrained case and to handle the limitations set for the control signal afterwards. GPC gives a closed formula to calculate the optimal control sequence. GPC can be transformed into a two degree of freedom polynomial structure. For the case of stable plants this polynomial structure can be transformed into internal model control (IMC) structure as well. An effective aposteriori constraint handling is shown applying the controller dynamics in the feedback of the static saturation.

D.W. Clarke (1987) has derived the GPC (Generalized Predictive Control) algorithm for unconstrained predictive control. This algorithm can be transformed to a 2DF so-called RST form, where R, S and T are polynomials of the shift operator (Camacho, 1998). The degrees and the coefficients of the polynomials depend on the tuning parameters of GPC.

In the paper this structure is converted to an IMC (Internal Model Control) structure. The IMC control algorithm is given for stable plants. Then an effective aposteriori constraint handling is provided.

3.1. Equivalent *GPC*, *RST* and *IMC* algorithms

A. The GPC algorithm

The plant is given by ARIMAX model

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + T(q^{-1})\frac{e(t)}{\Delta} \tag{5}$$

where $u(t)$ is the sampled input signal, $y(t)$ is the output signal, $e(t)$ is the zero mean value noise, d is the discrete dead time. Polynomials A, B and C are functions of the q^{-1} shift operator.

$$A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na}$$

$$B(q^{-1}) = b_1 q^{-1} + \ldots + b_{nb} q^{-nb}$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{nc} q^{-nc}$$

and

$$\Delta = 1 - q^{-1}$$

The cost function:

$$J = \sum_{j=n_{e1}}^{n_{e2}} \delta(j) \left[\hat{y}(t+j|t) - y_{ref}(t+j)\right]^2 + \sum_{j=1}^{n_u} \lambda_u(j) \left[\Delta u(t+j-1)\right]^2 \tag{6}$$

$\delta(j)$ and $\lambda(j)$ are weighting factors. Let us arrange them in diagonal matrices δ and $\lambda_{\mathbf{u}}$. By minimizing the cost function the following control rule is obtained:

$$\Delta\mathbf{u} = \left[\mathbf{H_\Delta}^T \delta \mathbf{H_\Delta} + \lambda_{\mathbf{u}}\right]^{-1} \mathbf{H_\Delta}^T \delta \left(\mathbf{y_{ref}} - \hat{\mathbf{y}}_{\mathbf{free}}\right) \tag{7}$$

Here matrix $\mathbf{H_\Delta}$ is a lower triangle matrix containing the points of the discrete step response. $\hat{\mathbf{y}}_{\mathbf{free}}$ gives the free response of the process, giving the effect of the past inputs to the system responses in the future points, supposing that the input signal is frozen at point $t-1$. If δ is a unity matrix and $\lambda_{\mathbf{u}}$ is a constant, then

$$\Delta\mathbf{u} = \left[\mathbf{H_\Delta}^T \mathbf{H_\Delta} + \lambda_u \mathbf{I}\right]^{-1} \mathbf{H_\Delta}^T \left(\mathbf{y_{ref}} - \hat{\mathbf{y}}_{\mathbf{free}}\right) \tag{8}$$

With the receding horizon strategy only the first element of $\Delta\mathbf{u}$ is applied as process input, and in the next sampling point the procedure is repeated (receding horizon strategy).

$$\Delta u(t) = \mathbf{K}\left(\mathbf{y_{ref}} - \hat{\mathbf{y}}_{\mathbf{free}}\right) = \sum_{i=N_1}^{N_2} k_i \left[y_{ref}(t+i) - y_{free}(t+i)\right] \tag{9}$$

where $\mathbf{K}$ is the first row of matrix $\left[\mathbf{H_\Delta}^T \mathbf{H_\Delta} + \lambda_u \mathbf{I}\right]^{-1} \mathbf{H_\Delta}^T$.

B. RST form of the GPC algorithm

The control algorithm is required in the following form:

$$R(q^{-1})\Delta u(t) = T(q^{-1})r(t) - S(q^{-1})y(t) \tag{10}$$

where R,S and T are the polynomials of the shift operator.

The control structure is shown in Fig. 4.

The RST structure is obtained from the GPC algorithm as follows (Camacho, 1998).

Let us solve the Diophantine equation below:

$$T(q^{-1}) = E_j(q^{-1})\Delta A(q^{-1}) + q^{-j} F_j(q^{-1}) \tag{11}$$

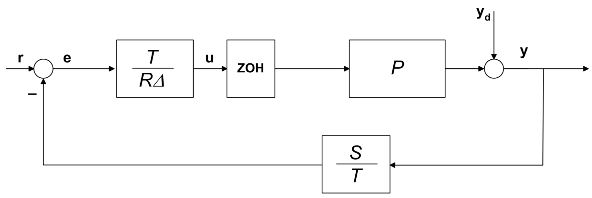


Figure 4. 2DF RST control structure

The solution provides polynomials E_j and F_j.

Then a second Diophantine equation is to be solved.

$$E_j(q^{-1})B(q^{-1}) = H_j(q^{-1})T(q^{-1}) + q^{-j}I_j(q^{-1}) \tag{12}$$

The solution gives polynomials H_j and I_j. Polynomials R and S are calculated by the following relationships:

$$R(q^{-1}) = \frac{T(q^{-1}) + q^{-1}\sum_{i=N_1}^{N_2} k_i I_i}{\sum_{i=N_1}^{N2} k_i} \tag{13}$$

$$S(q^{-1}) = \frac{\sum_{i=N_1}^{N2} k_i F_i}{\sum_{i=N_1}^{N2} k_i} \tag{14}$$

One advantage of this form is that the controllers can be given in advance. On the other hand the characteristic equation can be given and the stability can be checked based on its roots.

C. IMC form of the GPC algorithm

The RST structure can be converted to IMC (Internal Model Control) structure according to Fig. 5.

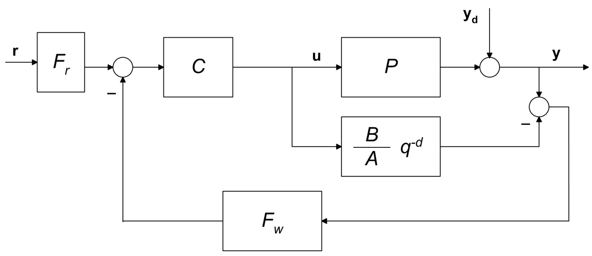


Figure 5. 2DF IMC control structure

If there is no disturbance and no plant-model mismatch, the performance of the control system between the reference signal r and the output signal y is determined by the forward path. In case of disturbance or plant-model mismatch the feedback becomes active to reject the effect of the disturbance. With filters F_r and F_w the dynamics of tracking and disturbance rejection will be different.

The IMC and the RST structure will be equivalent, if the controller C in the IMC structure (Fig. 5.) is chosen according to the following relationship.

$$C(q^{-1}) = \frac{S(q^{-1})A(q^{-1})}{F_w(q^{-1})\left(R(q^{-1})\Delta A(q^{-1}) + S(q^{-1})B(q^{-1})q^{-d}\right)} \tag{15}$$

$$\frac{T(q^{-1})}{S(q^{-1})} = \frac{F_r(q^{-1})}{F_w(q^{-1})} \tag{16}$$

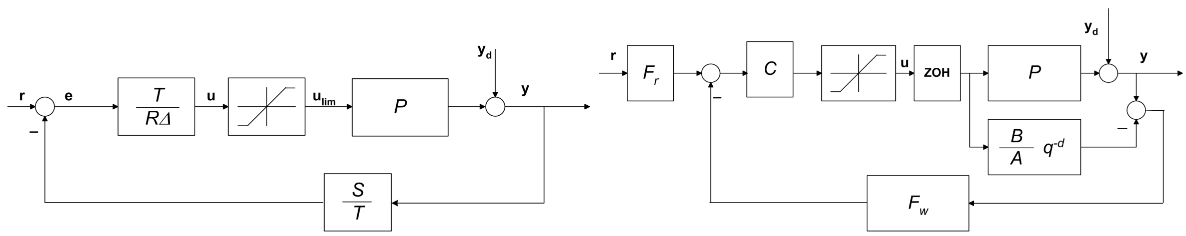
The simplest choice if $F_r = T$, and $F_w = S$.

The controller C can be considered as a quasi-inverse of the process. The characteristic equation is the denominator of controller C.

3.2. HANDLING OF CONSTRAINTS

In practical applications the control signal can change between its lower and upper limits. Saturation besides opening the control circuit and modifying the transient performance may cause integrator windup phenomenon and limit cycles.

In polynomial RST or IMC predictive control structures saturation has to be handled. Fig. 6(a) shows the simple saturation in the RST structure, while Fig. 6(b) shows saturation effect in the IMC circuit.



(a) RST (b) IMC

Figure 6. RST and IMC structure with saturation

IMC structure in this respect is expected to behave better, as the model of the plant is saturated in the same way as the plant itself.

The behavior of the IMC structure can be improved, if the whole dynamics of the controller is put in the feedback of the static nonlinear characteristics (Fig. 7), (Barta, 2005) . The controller in the feedback of the saturation is obtained by

$$C_{lim}(q^{-1}) = \frac{C(q^{-1}) - 1}{C(q^{-1})} \tag{17}$$

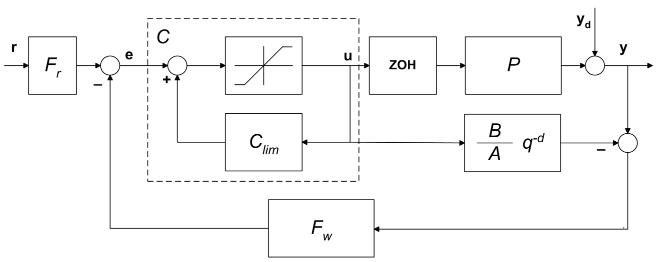


Figure 7. IMC structure with the controller dynamics in the feedback of the saturation dinamikja

In case of the same degree of the numerator and denominator in C_{lim}, an algebraic loop is obtained. Separating the constant part of the controller and including it in the slope of the saturation characteristics this problem is solved.

Example

Let us consider a simple example. The plant is a first order lag with transfer function $P(s) = \frac{1}{1+s}$. The tuning parameters of the predictive controller are:

- $N_1 = 1$
- $N_2 = 3$
- $N_u = 1$
- $\lambda_u = 0$

The parameters of the RST structure are:

- $R = 0.2037$
- $S = 2.9366 - 1.9366q^{-1}$
- $T = 1$

The parameters of the IMC structure:

- $F_r = 1$
- $F_w = 2.9366 - 1.9366q^{-1}$
- $C = \frac{4.9098-4.443q^{-1}}{1-0.5328q^{-1}}$

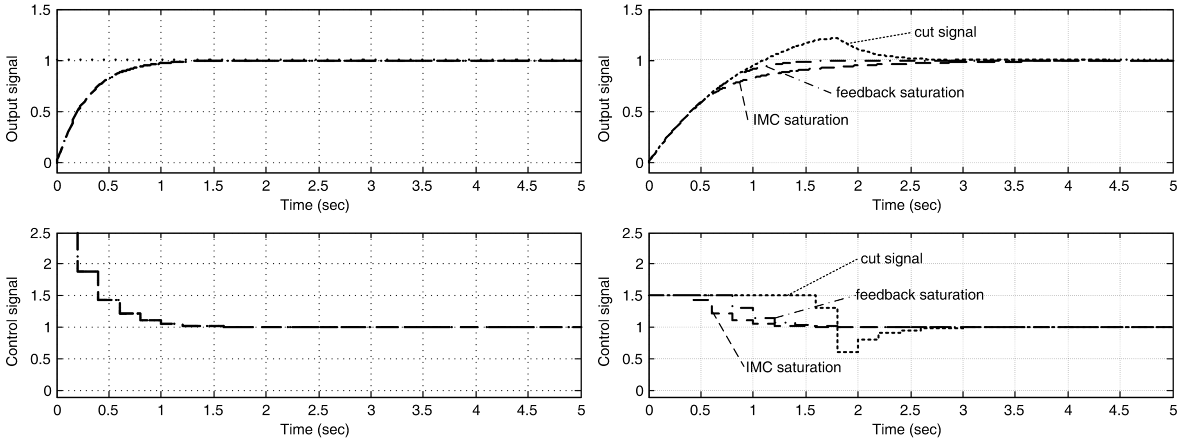
The pulse transfer function of the controller in the feedback of the saturation:

$$C_{lim}(z) = \frac{0.7963q - 0.7963}{q - 0.9048}$$

Fig. 8. shows the output and the control signals for step reference input without and with saturation of value ± 2.5. A unit step disturbance is acting at point $10sec$. It is seen that the feedback saturation gives the most favorable behavior.

4. SUMMARY

Optimisation considering constraints can be a time consuming operation especially in case of long prediction horizon. An important practical problem is decreasing the calculations executed between the sampling points in real time applications. The paper presents some possibilities to reduce the computation time. The calculations can be decreased significantly by using the coincidence points in the prediction horizon and blocking technique relating the control (manipulated variable) horizon. It is shown that by an optimal choice of the location of the points considered in both horizons the computation effort can be drastically decreased. Another approach is using the Generalized Predictive Control (GPC) algorithm for unconstrained case and to handle the limitations set for the control signal afterwards. For the case of stable plants GPC is transformed into internal model control (IMC) structure. An effective aposteriori constraint handling is shown applying the controller dynamics in the feedback of the static saturation.



(a) without saturation (b) with saturation

Figure 8. RST and IMC control of a first order lag system without and with saturation

ACKNOWLEDGEMENTS

The work has been supported by the University of Applied Science Cologne in the program of "Advanced Process Identification for Predictive Control" and by the program of EU-Socrates-Erasmus. The work of the authors from the BME was also supported by the fund of the Hungarian Academy of Sciences for control research and partly by the OTKA fund T029815. All supports are kindly acknowledged.

REFERENCES

1. Camacho, E.F., Bordons, C. (1998). Model Predictive Control, Springer Verlag, London.
2. Clarke, D.W., C. Mohtadi, P.S. Tuffs (1987). Generalized predictive control - Part I. The basic algorithm. Automatica, 24, 2, 137-148.
3. Haber, R., U. Schmitz, R. Bars (2004). Optimal choice of horizons for predictive control by using genetic algorithm. Annual Reviews in Control, Vol. 28, Issue 1, 53-58.
4. Qin, S.J., T.A. Badgwell (2003). A survey of industrial model predictive control technology. Control Engineering Practice, 11, pp. 733-764.
5. Schmitz, U., R. Haber, R. Bars (2003). Reduction of the optimization efforts with predictive control by using non-equidistant prediction points. 12th IFAC Workshop on Control Applications of Optimisation CAO'2003, Visegrad, Hungary
6. Barta, A., R. Bars, I. Vajk, Zs. Preitl (2004). Practical controller design considering input saturation, Carpathian Control Conference, Miskolc-Lillafured, Hungary, May 24-27, 2005.
7. Preitl, Zs., R. Bars, R., I. Vajk, R. Haber (2004). Internal Model representation for Generalized Predictive Control with constraint handling, IEEE 4th International Conference on Intelligent Systems Design and Application ISDA, Budapest, Hungary
8. Sekaj, I. (2002). Genetic Algorithms with Changing Criterion Functions, Intelligent Technologies - Theory and Applications, P.Sincak et al. (Eds), IOS Press, 2002, ISBN 1 58603 256 9, pp. 183-188